**ITEC 2150 Test 1**

**Question 1 (33 points)**

Modify the **Numbers.java** file to implement the proper exception handling and add the correct processing. Create a new exception, StringNotANumberException. Your NumberTester should catch this exception and determine which string is not a number and pass 0 back to the add method for it. If both strings are not a number, the sum should be 0. Your number tester should print the sum of each call. Your NumberTester must pass both String and int arguments to the Number class so that your exception will be exercised.

Hints: Use a string when you throw your StringNotANumberException to help you determine which parameter is not a number. Add a method that will loop around your call to add() until you have valid values.

Example output:

15 + 10 = 25

First is invalid

a + 10 = 10

First is invalid

Second is invalid

a + b = 0

Second is invalid

15 + b = 15

Submit the files **StringNotANumberException.java**, **NumbersTester.java** and **Numbers.java**

**Question 2 (33 points)**

You are creating a game. An **Entity** class is a class that describes a character in a game. An Entity has a **name**, **HP** (health points) and an **inventory** of items that are carried by the character. The Entity class also comes with useful methods to change the HP and the items carried by the character.

Create two classes that inherit from the Entity class.

**PlayableCharacter**

1. Create an appropriate constructor.

2. Override the addItem() method so that an item that is already carried by the character is not added to the inventory.

3. Override the removeItem() method so that you do not remove an item that is not in the inventory.

4. Override the takeDamage() method so that a random number decides whether the damage takes place or not. The user has a 20% chance of not taking any damage.

5. Implement the getGreeting() method so that the String "My life for Aiur!" is returned.

**NonPlayableCharacter**

1. Create an appropriate constructor.

2. Override the takeDamage() method so that no damage takes place.

3. Override the takeHealing() method so that no healing takes place.

4. Implement the getGreeting() method so that the String "Greetings! At your service." is returned.

GameTester

1. Create an object of PlayableCharacter and NonPlayableCharacter
2. Create inventory items that can be added to or removed from the player’s backpack.
   1. You should create both weapons and non-weapons.
3. Add items to both your Playable and NonPlayableCharater
   1. You must add the same item to the Playable more than once and remove an item that is not in the Playable objects backpack.
4. Have the player and non-player take damage and then call toString() on each.
5. Have the planer and non-player receive healing and then call toString() on each.

You are not allowed to modify Entity.java or Item.java.

Example output:

My life for Aiur!

Fred with 100 of health has the following items:

Sword true

Helmet false

Laptop true

Greetings! At your service.

Sally with 100 of health has the following items:

Sword true

Sword true

Helmet false

Laptop true

Taking Damage of 1:

Playable: Fred 100

NonPlayable: Sally 100

Taking Healing of 5:

Playable: Fred 105

NonPlayable: Sally 100

Taking Healing of 7:

Playable: Fred 112

NonPlayable: Sally 100

Taking Damage of 2:

Playable: Fred 110

NonPlayable: Sally 100

Taking Damage of 11:

Playable: Fred 99

NonPlayable: Sally 100

Taking Damage of 4:

Playable: Fred 95

NonPlayable: Sally 100

Taking Healing of 3:

Playable: Fred 98

NonPlayable: Sally 100

Taking Damage of 14:

Playable: Fred 84

NonPlayable: Sally 100

Taking Damage of 17:

Playable: Fred 67

NonPlayable: Sally 100

Taking Healing of 4:

Playable: Fred 71

NonPlayable: Sally 100

Submit the files **PlayableCharacter.java, NonPlayableCharacter.java and GameTester.java files**

**Question 3 (34 points)**

Create a program that takes a text file which contains integers and strings and outputs only the integers to the console. For example, if your input file is the following:

We all 200 live 399 in 455 a yellow

500

600

Submarine

We 100 all 3223 live in a 2356 yellow

23 submarine 123

The output to the console **and** your output file should be:

200 399 455 500 600 100 3223 2356 23 123

The integers are guaranteed to delimited by spaces.

Please name your class **FileProcess.java**. A sample text file, TestFile.txt is provided.